

Exo 1

1) Rech_Rapide (Tab, Mot: Str): Bool

VAR TRouv: Bool
J: Entier ; P: Ptr (Tmaillon)

Debut

TRouv ← Faux ;
j ← char.To.int (Mot[0]) - 64
P ← Tab [j].Tête ;

TQ (TRouv) et (P <> NULL) faire
 Si (Valeur (P) == Mot) Alors
 TRouv ← VRAI
 Fsi
 P ← suiv (P) ;

FTQ

FIN.

2) Supp_Rapide (Tab, mot: str)

VAR TRouv: Bool
J: Entier
P, T, Ptr (Tmaillon)

Debut

j ← char.To.int (Mot[0]) - 64
P ← Tab [j].Tête
TRouv ← Faux.

Si (Valeur (P) == Mot) Alors
 Tab [j].Tête ← suiv (P)
 liberer (P)

Sinon

TQ (TRouv) et (P <> NULL)
 T ← P
 P ← suiv (P)
 Si (Valeur (P) == Mot) Alors
 TRouv ← VRAI
 Fsi

FTQ

Si (TRouv) Alors
 Aff.adr (T, suiv (P))
 liberer (P) ;

Fsi

FIN.

VAR J: Entier
Q, Pos_Min, P, T: Ptr (Tmaillon)

Debut
j ← char.To.int (Letter) - 64
P ← Tab [j].Tête

TQ (suiv (P) <> NULL) faire

 Pos_Min ← P ;
 T ← suiv (P) ;

 TQ (T <> NULL) faire
 Compare
 Si (Valeur (T) < Valeur (Pos_Min))

 Pos_Min ← T ;
 Fsi
 T ← suiv (T) ;

 FTQ

 Si (P <> Pos_Min) Alors
 Permut (P, Pos_Min)
 Fsi

FTQ

FIN.

Permut (P: Ptr, Q: Ptr)

 var aux: Entier
 Debut

 aux ← Valeur (Q)

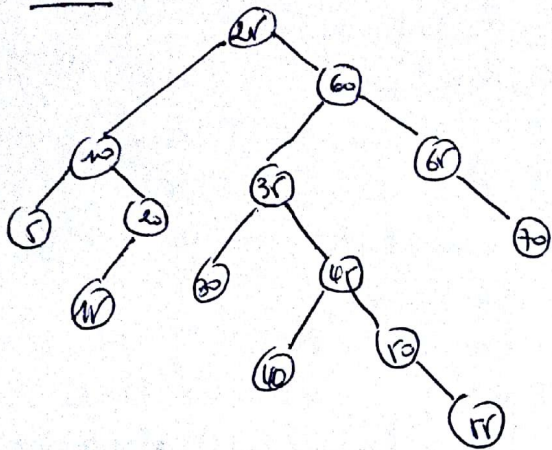
 Aff.val (Q, Valeur (P))

 Aff.val (P, aux)

FIN.

الترتيب التبادلي
الترتيب الأول
من أجل التبادلي

Exo 2



• Préfixe GDP :

20, 10, 5, 20, 15, 60, 30, 20, 40, 60, 50, 30, 65, 70

• Infixe GDP :

5, 10, 15, 20, 20, 30, 30, 40, 40, 50, 30, 60, 65, 70

• Postfixe GDP :

5, 15, 20, 10, 20, 30, 40, 50, 50, 40, 30, 65, 70, 60

• Supp-Père (Racine, a: Entier);

Re-Recherche-Père (Racine; N: Entier)

si (R == NULL) Alors

Supp-ARB (Racine, P)

Père

Fin

(*) la fonction Supp-ARB est la ^{me} fonction sur un TD.

(**) la fonction Recherche-Père retourne le père du N.

Exo 3

A) $O(1)$.

B) $O(\frac{1}{2}m + \frac{1}{20}n^2) = O(n+m)$

C) $O(\log_7 n)$;